

FASTER COMPARISON OF STOPPING TIMES BY NESTED CONDITIONAL MONTE CARLO

FABIAN DICKMANN AND NIKOLAUS SCHWEIZER

ABSTRACT. We show that deliberately introducing a nested simulation stage can lead to significant variance reductions when comparing two stopping times by Monte Carlo. We derive the optimal number of nested simulations and prove that the algorithm is remarkably robust to misspecifications of this number. The method is applied to several problems related to Bermudan/American options. In these applications, our method allows to substantially increase the efficiency of other variance reduction techniques, namely, Quasi-Control Variates and Multi-level Monte Carlo.

1. INTRODUCTION

In this paper, we propose a novel method for efficiently comparing the performance of different stopping times, i.e. we are interested in computing $\Delta = E[X_{\tau^A} - X_{\tau^B}]$ where X is a stochastic process in discrete time and τ^A and τ^B are two stopping times. A simple Monte Carlo algorithm for this problem consists of simulating N trajectories of X until both stopping times have occurred and then to average over the resulting N realizations of $X_{\tau^A} - X_{\tau^B}$. If τ^A and τ^B are similar, e.g., because they are two approximations of the same intractable stopping time, or solutions to two similar problems, this method tends to be inefficient, because sizeable contributions to Δ come only from the few regions in state space where τ^A and τ^B disagree.

Instead, we write

$$(1) \quad \Delta = E[E[X_{\tau^A} - X_{\tau^B} | \mathcal{F}_{\tau^\wedge}]], \quad \text{where} \quad \tau^\wedge = \min(\tau^A, \tau^B),$$

where $\mathcal{F}_{\tau^\wedge}$ denotes the information generated until the first of the stopping times occurs and propose the following two-stage simulation procedure: Simulate N trajectories of X until the first stopping time occurs, i.e., until τ^\wedge . Then simulate R conditionally independent copies of each of the N trajectories until the second stopping time $\tau^\vee = \max(\tau^A, \tau^B)$ and estimate Δ by the mean of the $R \cdot N$ realizations of $X_{\tau^A} - X_{\tau^B}$. The resulting estimator can be interpreted as estimating first for each of the N initial trajectories the inner conditional expectation in (1) by the mean over the R replications of that trajectory, and then averaging over the N initial trajectories to estimate the outer expectation.

Date: January 2014.

Key words and phrases. American Options, Bermudan Options, Branching, Importance Sampling, Multilevel Monte Carlo, Nested Simulation, Optimal Stopping, Splitting, Variance Reduction.

Both authors gratefully acknowledge financial support by the Deutsche Forschungsgemeinschaft through SPP 1324.

The idea of using subsimulations to estimate an inner conditional expectation relates our approach to the literature on Nested Simulation [14, 6]. In the applications considered there, there is a non-linear dependence on the inner conditional expectation so that the inner simulations are indispensable – and are generally regarded as an unavoidable burden. From this perspective, it is interesting to see that in our numerical examples, where inner simulations are introduced deliberately as a variance reduction technique, the estimated optimal numbers of inner paths, do not differ much from what is typically used in these applications, e.g. $R = 100$. Variance reduction by deliberately inserting a conditional expectation is a common technique if these conditional expectations are available in closed form. This classical method is known as Rao-Blackwellization or Conditional Monte Carlo [5, 2]. It is usually not applicable in our setting since closed-form expressions for expectations of stopped processes are rare. Since our method mimics Conditional Monte Carlo by Nested Simulation, we refer to it as *Nested Conditional Monte Carlo*.

We apply our method to three problems related to Bermudan option pricing.¹ In the first application, there is a genuine interest in comparing different stopping times: We consider the impact of parameter uncertainty on the performance of estimated optimal exercise strategies. In the two other applications, the true quantity of interest is $E[X_{\tau^A}]$ while X_{τ^B} serves as a control variate. Our method is used to enhance the efficiency of this control variate. For two variance reduction methods of this type, a Quasi-Control Variate as introduced in [9] and the Multilevel algorithm of [4], we demonstrate that Nested Conditional Monte Carlo can lead to sizeable efficiency gains. In fact, for both algorithms the additional variance reduction through incorporating our method is at least as large as the variance reduction of the original algorithm.

In a sense, our algorithm closely resembles the splitting algorithms for rare event simulation studied, e.g., in [20, 13]. In the applications considered in this literature (e.g. barrier option pricing or estimating the probability of large losses), the rare event typically consists of X taking exceptionally large or small values. Thus, the trigger events for replicating a trajectory are chosen as the hitting times of some threshold value of X . In this way, computational effort can be allocated efficiently to the regions where it is needed the most.

Our rare event, a large discrepancy between X_{τ^A} and X_{τ^B} , does not have such a nice structure, i.e., it is not easy to connect it a priori to particular values of X which might serve as a trigger for replications. Our key observation is that this type of knowledge is not necessary here: We simply start replications at an event, the first stopping time, which occurs once on *every* trajectory. An efficient allocation of computational effort to critical trajectories occurs endogenously by the following reasoning: $X_{\tau^A} - X_{\tau^B}$ is expensive to simulate if τ^A and τ^B lie far apart. Yet those cases where τ^A and τ^B lie far apart also carry a substantial probability for large values of $X_{\tau^A} - X_{\tau^B}$. In addition, in many cases the difference $X_{\tau^A} - X_{\tau^B}$ will not

¹ We refer to the options as Bermudan, since the sets of exercise times are finite. We could also interpret them as a time discretizations of American options.

depend strongly on what happens before τ^\wedge . Thus, we gain efficiency by shifting computational effort from the time interval $[0, \tau^\wedge]$ to the interval $[\tau^\wedge, \tau^\vee]$ (which is typically much shorter). To sum up, we use splitting with the objective of identifying important regions in time rather than in space – although, of course, the two cannot fully be disentangled. On a more abstract level, our results demonstrate that ideas from rare event simulation can be used to boost the efficiency of control variates even when dealing with “ordinary” events. As we will see in the numerical examples, this gives a highly efficient Monte Carlo algorithm which has only one free parameter, the number of replications R .

A particular advantage of splitting methods such as ours is that they result in unbiased estimators. This sets them apart from related algorithms such as Importance Sampling and particle methods (see the survey [8]) which also aim at an endogenous and efficient distribution of simulation costs. Unbiasedness is of particular importance in option pricing applications, where one wishes to calculate estimators which are known to have a positive or negative bias in order to construct confidence intervals, see e.g. [1].

The paper is organized as follows: Section 2 introduces the setting and the algorithm and derives a formula for the variance of the estimator. Section 3 characterizes situations where our algorithm leads to significant improvements over simple Monte Carlo and derives a formula for the optimal number of replications R . Moreover, we prove that the performance of the estimator is rather insensitive to moderate misspecifications of R : As long as R is, say, within 20% of the optimal value we achieve more than 99% of the optimal variance reduction. Section 4 demonstrates the algorithm’s efficiency in the three option pricing applications sketched above. Section 5 concludes. All proofs are in the appendix.

2. THE ALGORITHM

Consider a square-integrable, adapted, real-valued stochastic process $(X_j)_j$ on a complete filtered probability space $(\Omega, \mathcal{F}, (\mathcal{F}_j)_{j=0}^J, P)$ over the discrete time horizon $\{0, 1, \dots, J\}$. There are two stopping times τ^A and τ^B and we are interested in computing

$$\Delta = E[X_{\tau^A} - X_{\tau^B}]$$

by a Monte Carlo approach. We define the stopping times $\tau^\wedge = \min(\tau^A, \tau^B)$ and $\tau^\vee = \max(\tau^A, \tau^B)$ and the random variable $S = \text{sign}(\tau^A - \tau^B)$ and note that

$$\tau^A - \tau^B = S(\tau^\vee - \tau^\wedge), \quad \text{that} \quad X_{\tau^A} - X_{\tau^B} = S(X_{\tau^\vee} - X_{\tau^\wedge})$$

and that S is observable at time τ^\wedge . For our Monte Carlo approach, we assume that (conditionally) independent copies of random variables are available as needed on our probability space and propose the following two-stage simulation algorithm which is determined by two integer-valued, positive parameters N and R :

- A1. Simulate independent copies $X_0^{(i)}, \dots, X_{\tau^\wedge, (i)}^{(i)}$ of $X_0, \dots, X_{\tau^\wedge}$ for $i = 1, \dots, N$. Denote by $\mathcal{F}^{\tau^\wedge, (i)}$ the information generated along the i^{th} trajectory and by $S^{(i)}$ the associated copy of S .

- A2. Conditionally on $\mathcal{F}^{\tau^\wedge, (i)}$ simulate for each i with $S^{(i)} \neq 0$ and for $r = 1, \dots, R$ copies $X_{\tau^\wedge, (i)+1}^{(i,r)}, \dots, X_{\tau^\vee, (i,r)}^{(i,r)}$ of $X_{\tau^\wedge+1}, \dots, X_{\tau^\vee}$ which are independent across the i and conditionally independent across the r . If $S^{(i)} = 0$ and thus $\tau^\vee, (i,r) = \tau^\wedge, (i)$ set $X_{\tau^\vee, (i,r)}^{(i,r)} = X_{\tau^\wedge, (i)}^{(i)}$.
- A3. Estimate Δ by

$$(2) \quad \Delta^{(N,R)} = \frac{1}{N} \sum_{i=1}^N \frac{1}{R} \sum_{r=1}^R S^{(i)} (X_{\tau^\vee, (i,r)}^{(i,r)} - X_{\tau^\wedge, (i)}^{(i)})$$

In short, we simulate N independent trajectories of X until the first stopping time τ^\wedge . From τ^\wedge on, we simulate R copies of each trajectory until the second stopping time τ^\vee . Δ is estimated by the mean $\Delta^{(N,R)}$ of the $R \cdot N$ (dependent) samples of $X_{\tau^A} - X_{\tau^B}$ obtained in this way.

Figure 1 illustrates the simulation procedure for an example where the two stopping times are given in the form of exercise boundaries, the blue and red curves in the figure. Whenever the process X crosses either boundary, the stopping time occurs. From the small amounts of red and blue in the picture, we observe that the subsampling in Step A2 only has to be carried out rarely – when the two stopping times differ, implying that the additional simulation costs from subsampling tends to be small.

As will become clear below, $\Delta^{(N,R)}$ can be interpreted as an approximation of a so-called Conditional Monte Carlo estimator where conditional expectations are replaced by nested simulations (subsamples). We thus refer to the algorithm as the *Nested Conditional Monte Carlo* algorithm. The following proposition shows that $\Delta^{(N,R)}$ is unbiased and gives an expression for its variance.

Proposition 2.1. *We have $E[\Delta^{(N,R)}] = \Delta$ and*

$$\text{Var}(\Delta^{(N,R)}) = \frac{v_1}{N} + \frac{v_2}{RN}$$

where

$$v_1 = \text{Var}(E[X_{\tau^A} - X_{\tau^B} | \mathcal{F}_{\tau^\wedge}]) \quad \text{and} \quad v_2 = E[\text{Var}(X_{\tau^A} - X_{\tau^B} | \mathcal{F}_{\tau^\wedge})].$$

The basic motivation for the algorithm is as follows: If τ^A and τ^B are not far apart, Step A2 of the algorithm is much cheaper computationally than Step A1. If they happen to coincide, Step A2 is for free. Therefore, large values of R are comparatively cheap. Moreover, if circumstances are favorable, namely, if the bulk of the variance in $X_{\tau^A} - X_{\tau^B}$ actually comes from what happens between τ^A and τ^B , i.e., if $v_1 \ll v_2$, the estimator will behave like an estimator with $R \cdot N$ rather than N samples.

We close this section by pointing out that $\Delta^{(N,R)}$ can be understood as an interpolation between two well-known Monte Carlo algorithms: $R = 1$ corresponds to a simple Monte Carlo estimator and $R = \infty$ corresponds to Conditional Monte Carlo. For $R = 1$, the algorithm collapses to a simple Monte Carlo estimator $\Delta^{MC} = \Delta^{(N,1)}$ of $X_{\tau^A} - X_{\tau^B}$ along N sample trajectories of $X_0, \dots, X_{\tau^\vee}$. Moreover, we have

$$\text{Var}(\Delta^{MC}) = \frac{\text{Var}(X_{\tau^A} - X_{\tau^B})}{N} = \frac{v_1 + v_2}{N}.$$

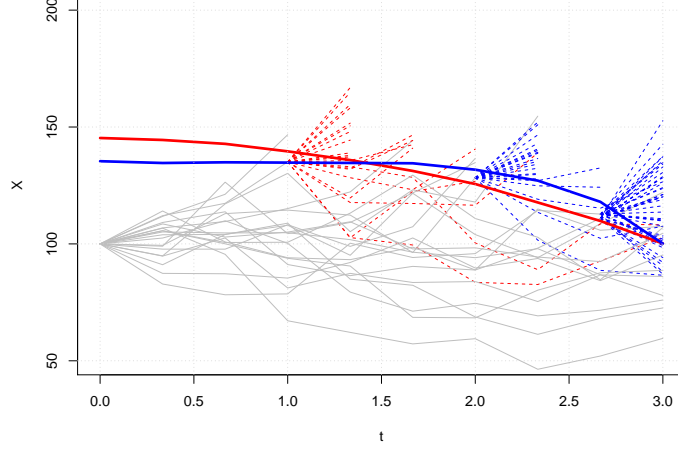


FIGURE 1. Simulated trajectories stopped and replicated at two exercise boundaries.

The final equality is the well-known conditional variance decomposition formula. The inner sum in the estimator $\Delta^{(N,R)}$,

$$\frac{1}{R} \sum_{r=1}^R S^{(i)}(X_{\tau^{\vee},(i,r)}^{(i,r)} - X_{\tau^{\wedge},(i)}^{(i)}),$$

can be interpreted as a Monte Carlo estimator for

$$D^{(i)} = E[S^{(i)}(X_{\tau^{\vee},(i,1)}^{(i,1)} - X_{\tau^{\wedge},(i)}^{(i)}) | \mathcal{F}_{\tau^{\wedge},(i)}].$$

which is exact in the limit $R \rightarrow \infty$. The limiting Monte Carlo estimator

$$\Delta^{CMC} = \frac{1}{N} \sum_{i=1}^N D^{(i)}$$

is the so-called Conditional Monte Carlo (CMC) estimator for Δ . In the applications we consider, the conditional expectation in $D^{(i)}$ typically cannot be computed explicitly and thus the estimator $\Delta^{CMC} = \Delta^{(N,\infty)}$ is purely of theoretical interest. The variance of Δ^{CMC} is given by

$$\text{Var}(\Delta^{CMC}) = \frac{v_1}{N},$$

thus reducing the variance by a factor $v_1/(v_1 + v_2)$ compared to Δ^{MC} . By employing nested simulation ($R > 1$) to approximate the conditional expectations, we construct implementable estimators which achieve at least part of this variance reduction.

3. CALIBRATING THE ALGORITHM

In the previous section, we saw that $\Delta^{(N,R)}$ achieves a guaranteed variance reduction compared to the simple Monte Carlo estimator $\Delta^{(N,1)}$. This comparison is however unfair since increasing the number of replications R leads to higher computational costs. In this section, we thus address the

following set of closely related questions: Given a fixed computational budget, when is it favorable to implement the estimator $\Delta^{(N,R)}$ with $R > 1$? How can we determine the optimal value of R ? How much do we gain compared to the simple Monte Carlo method with $R = 1$? How sensitive is the algorithm's performance to miscalibration?

It turns out that the answers to all these questions crucially depend on two natural conditions: Implementing $\Delta^{(N,R)}$ with $R > 1$ can lead to drastic computational efficiency gains 1) if the theoretical Conditional Monte Carlo estimator Δ^{CMC} would lead to a substantial variance reduction, and 2) if generating samples of $X_{\tau^\vee} - X_{\tau^\wedge}$ conditionally on X_{τ^\wedge} is cheaper than sampling copies of X_{τ^\wedge} , i.e., if the cost of a single sample is smaller in Step A2 of the algorithm than in Step A1. If τ^A and τ^B are sufficiently similar, the second condition is likely to hold, i.e., the difference between the stopping times will be much smaller in expectation than their minimum.

Observe next that the computational costs of implementing the estimator for fixed N and R is itself random since the number of time steps that have to be simulated depends on the realizations of the stopping times. This is the case even for the simple Monte Carlo estimator with $R = 1$. The following analysis is thus based on expected computational costs.

Denote by ρ_1 the expected computational cost of simulating a realization of X_{τ^\wedge} in Step A1 of the algorithm. ρ_1 takes into account the expected length $E[\tau^\wedge]$ of a path and the costs of evaluating both stopping times along that path. Denote by ρ_2 the expected computational cost of simulating a realization of $X_{\tau^\vee} - X_{\tau^\wedge}$ for given X_{τ^\wedge} in Step A2 of the algorithm. ρ_2 takes into account the expected length $E[\tau^\vee - \tau^\wedge]$ of such a path and the costs of evaluating either of the two stopping times along that path. The expected computational cost of implementing the estimator with parameters N and R is thus given by

$$c(N, R) = N\rho_1 + NR\rho_2.$$

For simplicity, we assume that v_1, v_2, ρ_1 and ρ_2 are strictly positive.

The next proposition characterizes how to optimally choose N and R for a given computational budget C . In particular, we derive an expression for the optimizer R^* which is independent of the overall budget, showing that in relative terms the optimal allocation of computational costs between Steps A1 and A2 is independent of C . For this reason, the question of calibrating the algorithm is basically reduced to finding a good choice of R . For the moment, we ignore the integer-constraints on N and R . We do however take into account that in order to obtain an implementable algorithm we must have $R \geq 1$. By identifying the situations where $R^* > 1$ we identify those cases where our Nested Conditional Monte Carlo algorithm is more efficient than simple Monte Carlo.

Proposition 3.1. *For any $C > 0$, the solution (N^*, R^*) to*

$$\min_{N, R} \text{Var}(\Delta^{(N,R)}) \quad \text{s.t.} \quad c(N, R) \leq C, \quad R \geq 1, \quad N \geq 0$$

is given as follows: If

$$(3) \quad \frac{\rho_1}{\rho_2} \frac{v_2}{v_1} > 1$$

then

$$R^* = \sqrt{\frac{\rho_1 v_2}{\rho_2 v_1}} \quad \text{and} \quad N^* = \frac{C}{\rho_1 + \rho_2 R^*}.$$

If condition (3) is violated, the optimal choice is

$$R^* = 1 \quad \text{and} \quad N^* = \frac{C}{\rho_1 + \rho_2}.$$

From condition (3) we can characterize the cases where the algorithm with $R > 1$ is preferable to simple Monte Carlo as follows: (3) is fulfilled if the cost of a single sample is smaller in Step A2 than in Step A1, $\rho_2 < \rho_1$, and if a perfect CMC estimator would reduce the variance by at least a factor 2, $v_1 < v_2$. If either of these conditions fails, (3) can only hold if the other condition is satisfied sufficiently strongly.

If N and R are such that the budget constraint $c(N, R) = C$ holds with equality then the variance of $\Delta^{(N, R)}$ can be written as

$$\text{Var}(\Delta^{(N, R)}) = \frac{V(R)}{C} \quad \text{where} \quad V(R) = (\rho_1 + \rho_2 R) \left(v_1 + \frac{v_2}{R} \right).$$

Therefore, in order to compare the resulting variance across different values of R it suffices to compare the values $V(R)$. The next proposition quantifies the gain from using our algorithm with R^* subsamples rather than a simple Monte Carlo estimator:

Proposition 3.2. *If condition (3) holds, the relative gain (variance reduction) from using Nested Conditional Monte Carlo with R^* subsamples instead of simple Monte Carlo is given by*

$$\gamma^* = \frac{V(R^*)}{V(1)} = \frac{\left(\sqrt{\frac{v_1}{v_2}} + \sqrt{\frac{\rho_2}{\rho_1}} \right)^2}{\left(1 + \frac{v_1}{v_2} \right) \left(1 + \frac{\rho_2}{\rho_1} \right)}.$$

Moreover,

$$\max \left(\frac{\rho_2}{\rho_1 + \rho_2}, \frac{v_1}{v_1 + v_2} \right) \leq \gamma^* \leq 4 \max \left(\frac{\rho_2}{\rho_1 + \rho_2}, \frac{v_1}{v_1 + v_2} \right).$$

The lower bound on γ^* shows that the variance parameters v_i and the cost parameters ρ_i independently place a bound on the variance reduction we can hope to achieve: We can reduce the variance at most by a factor $v_1/(v_1 + v_2)$, no matter how small ρ_2 is compared to ρ_1 . The intuitive reason for this is that $\Delta^{(N, R)}$ can never beat the theoretical CMC estimator Δ^{CMC} . Likewise, no matter how small the CMC-variance v_1 is compared to v_2 , we can never gain more than the speed-up from concentrating our simulations on the interval from τ^\wedge and τ^\vee instead of the whole interval from 0 to τ^\vee . This speed-up is captured by the ratio between ρ_2 and $\rho_1 + \rho_2$. Since our upper bound on γ^* is four times the lower bound, we see that the lower bound is never too far off. To sum up, we can expect drastic variance reductions if (and only if) $v_1 \ll v_2$ and $\rho_2 \ll \rho_1$.

In practical implementations, we will not be able to work with exactly R^* subsamples for at least two reasons: Since we will not know the parameters v_1, v_2, ρ_1 and ρ_2 , these have to be estimated in pilot simulations. Moreover, R has to be set to an integer value. Thus, it is important to make sure that

the performance of the algorithm is not too sensitive to the choice of R . The next proposition shows that this is indeed the case, giving an upper bound on the loss in variance reduction if we can only guarantee that R lies in an interval around R^* .

Proposition 3.3. *Suppose that $R^* > 1$ and $\alpha^{-1}R^* \leq R \leq \alpha R^*$ for some $\alpha > 1$. Then we have the following bound on the loss in variance reduction:*

$$\frac{V(R)}{V(R^*)} \leq \frac{1}{2} + \frac{\alpha + \alpha^{-1}}{4}, \quad \text{and thus} \quad \frac{V(R)}{V(1)} \leq \left(\frac{1}{2} + \frac{\alpha + \alpha^{-1}}{4} \right) \gamma^*.$$

This bound is fairly tight for realistic values of α . For $\alpha = 1.2$, implying that R is misspecified by about 20%, we are still within 1% of the optimal variance reduction. For $\alpha = 2$, almost 90 % of the optimal variance reduction are achieved. We thus conclude that even a crude attempt at optimizing the number of subsamples R should lead to near-optimal results.

A key observation in the proof of Proposition 3.3 is the identity $V(\alpha R^*) = V(\alpha^{-1}R^*)$. The next corollary collects some of its practical implications for the choice of R : If R^* is significantly larger than 1, then there is a wide interval of values for R which give an improvement over simple Monte Carlo: Any value of R which is smaller than the square of the optimum R^* is better than $R = 1$. Moreover, given a fixed computational budget it is always better to overestimate R^* by a fixed amount, than to underestimate it by the same amount. Finally, rounding R^* to the nearest integer can never produce an algorithm which is worse than simple Monte Carlo.

Corollary 3.4. *Suppose condition (3) holds, i.e., $R^* > 1$. Then the following assertion are true:*

- (i) *For every R with $1 < R < R^{*2}$ we have an improvement over simple Monte Carlo, $V(R) < V(1)$.*
- (ii) *Let $r > 0$ be such that $R^* - r \geq 1$. Then $V(R^* + r) < V(R^* - r)$.*
- (iii) *Let $R^\#$ be the integer nearest to R^* . If $R^\# > 1$, then $V(R^\#) < V(1)$.*

4. APPLICATIONS TO BERMUDAN OPTION PRICING

In this section, we illustrate our algorithm in a number of applications related to a well-known benchmark example [1, 7] from Bermudan option pricing, the valuation of a Bermudan max-call option in a Black-Scholes model with dividends. There are d stocks with price processes Y_t^d over the continuous time horizon $[0, T]$. Under the risk-neutral pricing measure, the stocks are independent, identically distributed geometric Brownian motions with volatility σ and drift $r - \delta$. Here, r is the risk-free interest rate and δ is the dividend yield. Assume that there is a finite, ordered set of exercise dates t_0, t_1, \dots, t_J in $[0, T]$ and write

$$X_j = e^{-r t_j} \left(\max_d Y_{t_j}^d - K \right)^+.$$

Thus, X_j is the discounted payoff from exercising a Bermudan max-call option with strike K at time t_j . The fair price at time 0 of this option is given by $E[X_{\tau^*}]$ where the optimal stopping time τ^* solves

$$\sup_{\tau} E[X_{\tau}].$$

Here, the supremum runs over all stopping times with values in $\{0, \dots, J\}$.

This stopping problem is numerically intricate unless the dimension d is small. One popular method for identifying confidence intervals for the fair price is the primal-dual approach put forward in [1]. First, one calculates an approximation τ of τ^* and estimates $E[X_\tau]$ which gives a lower bound due to the suboptimality of τ . Afterwards, τ can also be used to construct high-biased estimators, relying on the dual approach of [15, 18]. We focus here on the first step of approximating τ and evaluating the low-biased estimates.

Note that if we use Monte Carlo methods for both the construction of τ and the evaluation of $E[X_\tau]$, we need to use independent randomness in the two calculations to preserve the low-biasedness property, see the discussion in [12]. Throughout, we refer to the paths of Y used in the calculation of τ as training paths and to those in the estimation of $E[X_\tau]$ and related quantities as testing paths.

In our numerical implementations we focus on two well-established methods for calculating the approximately optimal stopping time τ , the least-squares Monte Carlo algorithm of Tsitsiklis and Van Roy [19] and the mesh method of Broadie and Glasserman [7]. In the implementation of the Tsitsiklis-Van Roy method, we use as basis functions at time j the monomials up to second-order in the stocks $Y_{t_j}^d$ and the payoff from immediate exercise X_j . For the mesh method, we follow the implementation in [4], including the use of control variates, and omit the details here.

Before we come to the applications, let us briefly discuss these two methods: Arguably, the greatest strength of least-squares Monte Carlo methods is that even generic implementations, like the one above, often achieve approximations within a few percent of the true value at very low computational costs. However, for each fixed choice of basis functions, the methods' bias can only be reduced down to some fixed level by increasing the number of training paths. There is typically no practicable, generic method for controlling the precision without substantial tuning of the algorithm (the choice of basis functions) and/or massive increases in computational effort.

We stick here to a fast, “vanilla” implementation of the algorithm which, in our view, has all the advantages of least-squares Monte Carlo. In particular, this implementation is well-suited as an easy-to-evaluate quasi-control variate in Section 4.2.²

In contrast, for the mesh method the bias can be controlled to arbitrary precision by increasing the number of training paths. No “clever idea”, e.g. a choice of basis functions, is necessary. The price to pay for this considerable advantage is that evaluating the stopping times is highly expensive: When using the mesh method, the bulk of the computational effort is taken up by calculating the realizations of τ along the simulated paths, since each path has to be compared with all training paths at each point in time. In fact, the examples of Sections 4.2 and 4.3 can both be interpreted as attempts

² For reasons of numerical performance, we abstain from using European prices as basis functions since these are expensive to evaluate in high dimensions. For the same reason, we use the Tsitsiklis-Van Roy method instead of the slightly more popular Longstaff-Schwartz algorithm [17]. When working with two sets of paths, training and testing, the two methods typically lead to almost indistinguishable results, see [3].

$\hat{\sigma} - \sigma$	0.005	0.01	0.015	0.02
$\Delta(\hat{\sigma})$	0.011	0.026	0.043	0.066
$E[X_{\tau^\sigma}]$	8.042	8.042	8.042	8.042
$E[X_{\tau^{\hat{\sigma}}}]$	8.031	8.016	7.999	7.976
$P(\tau^\sigma \neq \tau^{\hat{\sigma}})$	0.022	0.043	0.062	0.081
ρ_1	7.975	7.974	7.972	7.972
ρ_2	0.053	0.104	0.154	0.199
v_1	0.008	0.020	0.037	0.061
v_2	4.023	8.016	12.053	16.066
R^*	271.8	176.0	129.4	103.3
γ^*	0.016	0.026	0.037	0.047
speed-up	62.5	38.5	27.0	21.3

TABLE 1. Estimated simulation parameters for different values of $\hat{\sigma}$. The speed-up $1/\gamma^*$ from using Nested CMC is given in the last row.

at increasing the mesh method's applicability by developing generic and efficient control variates.

4.1. Assessing Parameter Uncertainty. In our first numerical example, there is a genuine interest in comparing two stopping times. We study the problem of estimating the sensitivity of stopping times estimated by the Tsitsiklis-Van Roy method to parameter misspecifications, namely, a misspecified volatility. Denote by τ^σ the stopping time calculated from training paths which have the correct volatility σ , while $\tau^{\hat{\sigma}}$ is calculated from training paths with volatility $\hat{\sigma} \neq \sigma$. We wish to estimate the costs of exercising the option based on such a misspecified calculation, i.e.,

$$\Delta(\hat{\sigma}) = E[X_{\tau^\sigma} - X_{\tau^{\hat{\sigma}}}]$$

where the expectation is, of course, taken with respect to the correct model with volatility σ .

Following an example in [1, 7], we assume 10 exercise dates $t_0, \dots, t_9 = T$ which are equally distributed over the time horizon $[0, T]$, and work with the following set of parameters: $d = 2$, $T = 3$, $r = 0.05$, $\delta = 0.1$, $\sigma = 0.2$, $K = 100$, $Y_0^d = 90$. We use 100000 training paths of the underlying Brownian motion for calculating the stopping times and keep these fixed throughout. Table 1 reports estimates of the parameters ρ_i and v_i for different values of the misspecified volatility $\hat{\sigma}$.

The first thing to observe from the table is that in all four cases Nested CMC leads to a substantial variance reduction, varying between a factor of about 60 and about 20, with the largest gains if σ and $\hat{\sigma}$ are most similar. The ratio between v_1 and v_2 is fairly constant and (much) smaller than the ratio between ρ_2 and ρ_1 which is thus decisive for the achieved variance

reduction. We also report the probability that the two stopping times differ – so that the subsimulations actually have to be carried out – and find that it lies between 2% and 8%. These numbers are one key reason for the small values of ρ_2 and the high optimal numbers of subsamples (between 103 and 272).

The units in which we report the ρ_i are irrelevant (only the ratios matter) and chosen deliberately (but fixed across the table). More importantly, it should be emphasized that the exact values of these numbers inevitably vary across different numerical implementations of the method. In the above table, we tried to estimate these parameters as accurately as possible. Yet, as was shown in Section 3, this is not necessary – rough estimates are sufficient in practice.

4.2. Improved Quasi-Control Variates. In this section and the next, we turn to the more classical problem of calculating $E[X_{\tau^A}]$ for a given stopping time τ^A . Introduce a second stopping time τ^B and write

$$E[X_{\tau^A}] = E[X_{\tau^B}] + E[X_{\tau^A} - X_{\tau^B}].$$

In a classical control variate approach, one would choose τ^B such that the first expected value on the right hand side can be calculated explicitly and would then estimate only the second one by Monte Carlo. A popular example is the choice $\tau^B = J$ which corresponds to using European options as control variates.³ In a more abstract setting, Emsermann and Simon [9] pointed out that it can sometimes be beneficial to work with so-called quasi-control variates, i.e., a control variate whose expected value is not known explicitly.⁴ This is the case if X_{τ^B} is significantly cheaper to simulate than X_{τ^A} . In that case, the first summand can be estimated with many (cheap) simulations of X_{τ^B} . For the second summand, only a small number of (expensive) paths may be necessary due to the variance reduction effect of X_{τ^B} . Finally, note that the second expectation is exactly the type of term which can be estimated efficiently by Nested CMC. Thus, we can hope to significantly enhance quasi-control variates by our method.

We retain the numerical example of the previous section, except that we increase the dimension to $d = 3$. We choose τ^A as an approximate optimal stopping time calculated by the mesh method with control variate from [4] with 2500 training paths. As τ^B we choose a Tsitsiklis-Van Roy stopping time with 100,000 training paths. In Table 2 we state estimates of the expected values of $\mu^A = E[X_{\tau^A}]$, the variance $v^A = \text{Var}(X_{\tau^A})$ and the cost ρ^A for generating a sample of X_{τ^A} , as well as the corresponding quantities for τ^B . As expected, v^A and v^B are similar, but ρ^A is by a factor *three thousand* larger than ρ^B . Note also that μ^A is considerably larger than μ^B .

³There is a slight subtlety here, depending on whether one uses X_J or $E[X_J|\mathcal{F}_{\tau^A}]$ as a control variate, the European payoff or the European price. The second, superior choice can be understood as a conditional Monte Carlo estimator with $R = \infty$. This yields, however, typically a larger value of v_1 than the control variates we consider since $\tau^B = J$ is not necessarily a good approximation of τ^* . Moreover, European prices are not always available in closed form.

⁴We are unaware of previous applications of Quasi-Control Variates in Bermudan/American option pricing. As the present example shows, they are reasonably generic and powerful variance reduction techniques for this important problem.

μ^A	v^A	ρ^A	μ^B	v^B	ρ^B	v_1	ρ_1	v_2	ρ_2
11.276	182	37.92	11.224	206	0.0124	0.044	36.23	19.536	1.728

TABLE 2. Estimated simulation parameters.

Since both estimates have a downward bias, this reflects the greater accuracy of the mesh method. In particular, μ^A lies within the 95% confidence interval $[11.265, 11.308]$ for $E[X_{\tau^*}]$ from [1].

Denote by $\rho(R)$ and $v(R)$ the computational costs and the variance per testing path when estimating $E[X_{\tau^A} - X_{\tau^B}]$ by Nested CMC with R replications, i.e.,

$$\rho(R) = \rho_1 + R \rho_2 \quad \text{and} \quad v(R) = v_1 + \frac{v_2}{R}$$

where the ρ_i and v_i are defined exactly as in Section 2. Let N^B be the number of paths used to estimate μ^B and let N be, as before, the number of paths used in the estimation of $\mu^A - \mu^B$. For a given computational budget C and fixed R , the optimal choice of N^B and N is given as the solution of

$$\min_{N^B, N} \frac{v^B}{N^B} + \frac{v(R)}{N} \quad \text{s.t.} \quad \rho^B N^B + \rho(R)N \leq C.$$

By a calculation similar to those in [9] (or the proof of our Proposition 3.1) it follows that the optimal ratio between N and N^B is given by

$$(4) \quad \frac{N^B}{N} = \sqrt{\frac{v^B}{v(R)} \frac{\rho(R)}{\rho^B}}.$$

regardless of the size of the computational budget. Finally, observe that the optimal value R^* of R is the same as in Section 3 regardless of how we allocate computational effort between the estimations of μ^B and $\mu^A - \mu^B$. From the values of the v_i and ρ_i we note that $R^* = 97.037$ and the estimated gain γ^* in the simulation of $\mu^A - \mu^B$ is given by $\gamma^* = 0.067$, corresponding to a speed-up of almost fifteen times in this part of the estimation.

Table 3 compares the performance of three Monte Carlo estimators for μ^A which have (approximately) the same computational costs and with parameters guided by the above considerations. The first line gives the variance of a direct Monte Carlo estimator of μ^A with 3,150 sample paths. The second line shows the variance of a simple quasi-control variate estimator ($R = 1$) with $N^B = 536,178$ paths in the estimation of μ^B and $N = 2,989$ paths in the estimation of $\mu^A - \mu^B$. The third line shows the variance of a quasi-control variate estimator with $R = 100$ replications in each of the $N = 468$ paths in the estimation of $\mu^A - \mu^B$ and $N^B = 1,784,813$ paths in the estimation of μ^B . We thus see an improvement of more than a factor 100, which comes in equal parts from the quasi-control variate and from including nested simulations.

In the present example, the ratio between v_1 and v_2 is far more favorable than the ratio between ρ_1 and ρ_2 , implying that the latter ratio governs the variance reduction we achieve. This is due to the relatively high value of ρ_1 which arises since in about 60% of cases it is the cheap Tsitsiklis-Van Roy

Method	Variance	Running time
Simple Monte Carlo	60.4×10^{-3}	125s
Quasi-Control Variate	6.77×10^{-3}	121s
Quasi-Control Variate with Nested CMC	0.514×10^{-3}	105s

TABLE 3. Comparison of the three methods with similar running times. This table reports averages over hundred runs of the simulation implemented in C++ on a standard system with a 2.6 GHz AMD processor.

stopping time which stops first. One can construct an even more efficient quasi-control variate by modifying the Tsitsiklis-Van Roy stopping time to be *slightly* biased towards late stopping, thus increasing the variance v_1 but decreasing ρ_2 . This can be achieved, e.g., by adding a small constant to the estimated continuation values.

4.3. An Improved Multilevel Algorithm. Multilevel Monte Carlo methods, initiated by Heinrich [16] and Giles [11], can easily be understood as an extension of the quasi-control variate approach: Instead of a single quasi-control variate, there is a sequence of random variables, where each element in the sequence serves as a quasi-control variate to its successor. In a recent paper, Belomestny, Dickmann and Nagapetyan [4] introduced and analyzed a multilevel method for our problem of numerically evaluating approximations of $E[X_{\tau^*}]$.

The algorithm of [4] can be summarized as follows: Fix a number of levels L and an increasing sequence k_0, \dots, k_L and consider the sequence of stopping times $(\tau(k_i))_i$ which are approximations of τ^* calculated by the mesh method with k_i training paths. Of these stopping times, $\tau(k_L)$ is both, the most accurate and the most expensive. Thus, in order to estimate $E[X_{\tau(k_L)}]$ we write

$$(5) \quad E[X_{\tau(k_L)}] = E[X_{\tau(k_0)}] + \sum_{i=1}^L E[X_{\tau(k_i)} - X_{\tau(k_{i-1})}]$$

In the multilevel method of [4], each of the summands on the right hand side is estimated independently by Monte Carlo with N_i testing paths, where the sequence N_0, \dots, N_L is *decreasing*. The motivation for the algorithm is as follows: $\tau(k_0)$ is cheap to evaluate and $E[X_{\tau(k_0)}]$ can thus be estimated with many testing paths. The summands in (5) become more expensive as i increases. Yet since their contribution to the overall estimate is comparatively small, one can afford to estimate them with fewer sample paths.

For this method (under a suitable choice of parameters), it was shown in [4] that the overall computational effort (training and testing) required for a mean squared error of ε behaves like $\varepsilon^{-2.5}$ as ε gets small. This is a significant improvement over a simple Monte Carlo method with ε^{-3} . In the following, we demonstrate that applying Nested Conditional Monte Carlo in the estimation of $E[X_{\tau(k_i)} - X_{\tau(k_{i-1})}]$ can lead to a substantial speed-up in

Level i	1	2
$E[X_{\tau(k_i)} - X_{\tau(k_{i-1})}]$	0.886	0.026
ρ_1	9.8	111.3
ρ_2	1.4	1.8
v_1	2.292	0.037
v_2	55.429	14.485
γ^*	0.28	0.031
R^*	13.018	158.707

TABLE 4. Simulation parameters at the two levels.

the non-asymptotic regime. For an entirely different application, the idea of using splitting techniques to speed-up multilevel Monte Carlo is mentioned as a possibility already in [11]. We are, however, unaware of later research which followed this suggestion.

We retain the numerical example of the two previous sections but increase the dimension to $d = 5$. We work with two levels, $L = 2$, and $(k_0, k_1, k_2) = (100, 1000, 10000)$. The three stopping times $\tau(k_i)$ are calculated by the mesh method with European control variate, specified exactly as in [4]. Besides the number of training paths k_i , we also increase the approximation quality of the numerical integration in the European control variate across levels, choosing precision parameters $(u_0, u_1, u_2) = (0.5, 0.05, 0.005)$, see [4] for details. The training paths in the construction of $\tau(k_0)$ and $\tau(k_1)$ are subsets of the training paths for the true target stopping time $\tau(k_2)$, so that the additional effort from working with three instead of one stopping times is negligible at the training stage. In light of (5), we can apply our Nested CMC twice, and obtain two sets of parameters ρ_i and v_i which are summarized in Table 4.

We thus see, that Nested CMC leads to drastic speed-up of about a factor 32 at the high-precision level $i = 2$, and to a still decent one of about 3.5 at the intermediate level $i = 1$.⁵ At the “base level”, i.e., the calculation of $E[X_{\tau(k_0)}] = 15.698$ we have a variance of $\text{Var}(X_{\tau(k_0)}) = 251.3$ and a cost per sample which we normalize to 1. Following (5), the expected value we are calculating is thus

$$E[X_{\tau(k_L)}] = 15.698 + 0.886 + 0.026 = 16.610$$

which is well within the confidence interval $[16.60, 16.66]$ for $E[X_{\tau^*}]$ from [1] for this example. In determining the optimal number of testing paths for estimating each summand in (5), we use the following generalization of (4) found, e.g. in [11, 4]: For a fixed computational budget, the *squared* number of paths N_i^2 in the estimation of each summand should be proportional to the variance divided by the costs per sample for this summand.

⁵This effect would become more pronounced if we included further levels of higher precision: These have an even greater speed-up factor from using Nested CMC.

Method				Variance
Simple Monte Carlo	N			0.13
	1790			
Multilevel	N_0	N_1	N_2	0.033
	38760	5550	880	
Multilevel with Nested CMC	86780	2650	100	0.0067

TABLE 5. Overall expected variances of the three methods with identical expected computational costs.

Table 5 compares Multilevel Monte Carlo with and without nesting for a fixed expected computational budget of 200000 time units. As suggested by Table 4, we use $R = 13$ and $R = 159$ replications in the Nested CMC algorithms at the two levels.

We also present results for a simple Monte Carlo estimator of the same expected value, $E[X_{\tau(k_2)}]$, under the same budget. Simple Monte Carlo has a cost per sample of 112.9 and $\text{Var}(X_{\tau(k_2)}) = 234.1$. There is a variance reduction by a factor 19.6 between simple Monte Carlo and Multilevel Monte Carlo with Nested CMC, the larger part of which (a factor 5) comes from incorporating the nested simulations.

5. CONCLUSION

In this paper, we have introduced Nested Conditional Monte Carlo, a simple Monte Carlo method for estimating the difference between two stopped versions of the same stochastic process. The algorithm is easy to calibrate by estimating two variance and two running-time parameters. Moreover, rough parameter estimates provably suffice for a near-optimal performance of the method. We demonstrated that besides its direct applications our method can be used as a generic tool for enhancing variance reduction methods for stopping problems. In fact, we can see little reason why one should implement the Quasi-Control Variate or Multilevel Monte Carlo methods of Section 4 without including subsimulations. The resulting variance reduction methods are very efficient and, unlike classical control variates, do not require that anything can be computed explicitly.

In terms of applications, we have focused on Bermudan option pricing but many other fields of application are conceivable. As examples, consider credit risk modelling, where events of default and distress are often modelled by stopping times, or pricing heuristics in revenue management as in [10] where the near-optimal timing of sales and promotions is studied.

In our algorithm, we use the same number of replications on each path. Yet, as demonstrated in a different type of application in [6], numerical efficiency can be improved by allocating more replications to “critical” trajectories. An extension of our method which achieves this – while retaining unbiasedness – splits the trajectories at *every* time point between τ^\wedge and τ^\vee . In this way, trajectories with a large value of $\tau^\vee - \tau^\wedge$ are automatically investigated more intensively. Alternatively – if one is not concerned about

a small bias – combinations of our method with Importance Sampling might be fruitful. We leave these and further extensions and applications of our method to future research.

APPENDIX A. PROOFS

Proof of Proposition 2.1. To see the unbiasedness, note that

$$\begin{aligned} E[\Delta^{(N,R)}] &= \frac{1}{N} \sum_{i=1}^N \frac{1}{R} \sum_{r=1}^R E[S^{(i)}(X_{\tau^{\vee},(i,r)}^{(i,r)} - X_{\tau^{\wedge},(i)}^{(i)})] \\ &= \frac{1}{N} \sum_{i=1}^N \frac{1}{R} \sum_{r=1}^R E[X_{\tau^A} - X_{\tau^B}] = \Delta \end{aligned}$$

where the second equality simply used that the term inside the expectation is an independent copy of $X_{\tau^A} - X_{\tau^B}$. For the variance, note first that the outer sum over i is a sum of independent, identically distributed random variables and thus

$$\text{Var}(\Delta^{(N,R)}) = \frac{1}{N} \text{Var} \left(\frac{1}{R} \sum_{r=1}^R S^{(1)}(X_{\tau^{\vee},(1,r)}^{(1,r)} - X_{\tau^{\wedge},(1)}^{(1)}) \right).$$

Applying the conditional variance decomposition formula yields

$$\text{Var}(\Delta^{(N,R)}) = \frac{v_1}{N} + \frac{v_2}{RN}$$

with

$$v_1 = \text{Var} \left(E \left[\frac{1}{R} \sum_{r=1}^R S^{(1)}(X_{\tau^{\vee},(1,r)}^{(1,r)} - X_{\tau^{\wedge},(1)}^{(1)}) \middle| \mathcal{F}^{\tau^{\wedge},(1)} \right] \right)$$

and

$$v_2 = R \cdot E \left[\text{Var} \left(\frac{1}{R} \sum_{r=1}^R S^{(1)}(X_{\tau^{\vee},(1,r)}^{(1,r)} - X_{\tau^{\wedge},(1)}^{(1)}) \middle| \mathcal{F}^{\tau^{\wedge},(1)} \right) \right].$$

and it remains to see that these values of v_1 and v_2 coincide with those in the proposition. Note that the summands are independent and identically distributed conditionally on $\mathcal{F}^{\tau^{\wedge},(1)}$. For v_1 this implies that

$$E \left[S^{(1)}(X_{\tau^{\vee},(1,r)}^{(1,r)} - X_{\tau^{\wedge},(1)}^{(1)}) \middle| \mathcal{F}^{\tau^{\wedge},(1)} \right]$$

does not depend on r and thus

$$v_1 = \text{Var} \left(E \left[S^{(1)}(X_{\tau^{\vee},(1,1)}^{(1,1)} - X_{\tau^{\wedge},(1)}^{(1)}) \middle| \mathcal{F}^{\tau^{\wedge},(1)} \right] \right).$$

A similar argument for v_2 now yields

$$v_2 = E \left[\text{Var} \left(S^{(1)}(X_{\tau^{\vee},(1,1)}^{(1,1)} - X_{\tau^{\wedge},(1)}^{(1)}) \middle| \mathcal{F}^{\tau^{\wedge},(1)} \right) \right].$$

Noting that $S^{(1)}(X_{\tau^{\vee},(1,1)}^{(1,1)} - X_{\tau^{\wedge},(1)}^{(1)})$ and $\mathcal{F}^{\tau^{\wedge},(1)}$ are copies of $X_{\tau^A} - X_{\tau^B}$ and $\mathcal{F}_{\tau^{\wedge}}$ allows to conclude the proof. Finally, let us emphasize that the above argument does take into account the fact that on *some* trajectories – those trajectories where τ^A and τ^B coincide – $X_{\tau^A} - X_{\tau^B}$ is $\mathcal{F}_{\tau^{\wedge}}$ measurable. \square

Proof of Proposition 3.1. Since the objective function decreases in both R and N , it is clear that the budget constraint holds with equality at the optimum, $c(N, R) = C$. Solving the constraint for N and substituting the result into the objective yields

$$\min_R \frac{1}{C} \left(v_1(\rho_1 + \rho_2 R) + v_2 \frac{\rho_1 + \rho_2 R}{R} \right) \quad \text{s.t.} \quad R \geq 1.$$

Using that the minimization problem is invariant to monotone transformations, we can write this as

$$\min_R \frac{v_1 \rho_2}{v_2 \rho_1} R + \frac{1}{R} \quad \text{s.t.} \quad R \geq 1.$$

Clearly, the solution to this convex minimization problem is $R^* = \max(1, R')$ where R' is the solution of the associated unconstrained minimization problem which is given by $R' = \sqrt{\frac{v_2 \rho_1}{v_1 \rho_2}}$. \square

Proof of Proposition 3.2. The formula for γ^* follows with a few algebraic manipulations after substituting $R^* = \sqrt{\frac{v_2 \rho_1}{v_1 \rho_2}}$ into V . We turn to the lower bound. By symmetry, it suffices to prove that for all positive real numbers a and b with $ab \leq 1$

$$\frac{(a+b)^2}{(1+a^2)(1+b^2)} \geq \frac{a^2}{1+a^2}.$$

To see this, note that we can bound the numerator as follows

$$(a+b)^2 \geq a^2 + ab \geq a^2 + a^2 b^2 = a^2(1+b^2).$$

For the upper bound it suffices to observe that

$$\begin{aligned} \frac{(a+b)^2}{(1+a^2)(1+b^2)} &\leq \frac{2a^2}{(1+a^2)(1+b^2)} + \frac{2b^2}{(1+a^2)(1+b^2)} \\ &\leq \frac{2a^2}{1+a^2} + \frac{2b^2}{1+b^2} \\ &\leq 4 \max \left(\frac{a^2}{1+a^2}, \frac{b^2}{1+b^2} \right). \end{aligned}$$

where we used in the first step that $(a+b)^2 \leq 2(a^2 + b^2)$. \square

Proof of Proposition 3.3. Observe first that we can write

$$V(R) = \rho_1 v_1 + \rho_2 v_2 + \rho_2 v_1 \left(R + \frac{R^{*2}}{R} \right)$$

and

$$V(\alpha R^*) = \rho_1 v_1 + \rho_2 v_2 + \rho_2 v_1 (\alpha + \alpha^{-1}) R^*.$$

Therefore we have $V(\alpha R^*) = V(\alpha^{-1} R^*)$ and by convexity $V(R) \leq V(\alpha R^*)$ for all R in the interval. It thus suffices to prove the upper bound for

$$\frac{V(\alpha R^*)}{V(R^*)} = \frac{\Gamma + \alpha + \alpha^{-1}}{\Gamma + 2} \quad \text{where} \quad \Gamma = \frac{\rho_1 v_1 + \rho_2 v_2}{\rho_2 v_1 R^*}.$$

Since $\alpha + \alpha^{-1} \geq 2$, we can bound this expression from above by replacing Γ with a smaller number. In particular, $\Gamma \geq 2$ yields the desired inequality

$$\frac{V(\alpha R^*)}{V(R^*)} \leq \frac{2 + \alpha + \alpha^{-1}}{4}.$$

To see that we indeed have $\Gamma \geq 2$, note that by inserting the expression for R^* we can write

$$\Gamma = \sqrt{\frac{\rho_2 v_2}{\rho_1 v_1}} + \sqrt{\frac{\rho_1 v_1}{\rho_2 v_2}}.$$

$\Gamma \geq 2$ now follows from the fact that $x + x^{-1} \geq 2$ for all $x \geq 0$. \square

Proof of Corollary 3.4. In the proof of Proposition 3.3 we saw that $V(\alpha R^*) = V(\alpha^{-1} R^*)$. For $\alpha = R^*$ this gives $V(R^{*2}) = V(1)$. Thus, (i) follows from the convexity of V . The argument for (ii) is similar. For (iii) note first that if $R^* < 1.5$ we have $R^\# = 1$ and nothing is to prove. By (i) it thus suffices to show $R^\# < R^{*2}$ for $R^* \geq 1.5$. To see this, note that $R^\# \leq R^* + \frac{1}{2} < R^{*2}$ where the last inequality holds for all $R^* > \frac{1+\sqrt{3}}{2} \approx 1.37$. \square

REFERENCES

- [1] Leif Andersen and Mark Broadie. Primal-dual simulation algorithm for pricing multidimensional american options. *Management Science*, 50(9):1222–1234, 2004.
- [2] Søren Asmussen and Klemens Binswanger. Simulation of ruin probabilities for subexponential claims. *Astin Bulletin*, 27(2):297–318, 1997.
- [3] Denis Belomestny. Pricing bermudan options using nonparametric regression: optimal rates of convergence for lower estimates. *Finance and Stochastics*, 15(4):655–683, 2011.
- [4] Denis Belomestny, Fabian Dickmann, and Tigran Nagapetyan. Pricing american options via multi-level approximation methods. *arXiv preprint arXiv:1303.1334*, 2013.
- [5] Phelim Boyle, Mark Broadie, and Paul Glasserman. Monte carlo methods for security pricing. *Journal of Economic Dynamics and Control*, 21(8):1267–1321, 1997.
- [6] Mark Broadie, Yiping Du, and Ciamac C. Moallemi. Efficient risk estimation via nested sequential simulation. *Management Science*, 57(6):1172–1194, 2011.
- [7] Mark Broadie and Paul Glasserman. A stochastic mesh method for pricing high-dimensional american options. *Journal of Computational Finance*, 7(4):35–72, 2004.
- [8] René Carmona, Pierre Del Moral, Peng Hu, and Nadia Oudjane. An introduction to particle methods with financial applications. In René A. Carmona, Pierre Del Moral, Peng Hu, and Nadia Oudjane, editors, *Numerical Methods in Finance*, volume 12 of *Springer Proceedings in Mathematics*, pages 3–49. Springer Berlin Heidelberg, 2012.
- [9] Markus Emsermann and Burton Simon. Improving simulation efficiency with quasi control variates. *Stochastic Models*, 18, 2002.
- [10] Youyi Feng and Guillermo Gallego. Optimal starting times for end-of-season sales and optimal stopping times for promotional fares. *Management Science*, 41(8):1371–1391, 1995.
- [11] Mike B. Giles. Multilevel monte carlo path simulation. *Operations Research*, 56(3):607–617, 2008.
- [12] Paul Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer, 2004.
- [13] Paul Glasserman, Philip Heidelberger, Perwez Shahabuddin, and Tim Zajic. Multi-level splitting for estimating rare event probabilities. *Operations Research*, 47(4):585–600, 1999.
- [14] Michael B. Gordy and Sandeep Juneja. Nested simulation in portfolio risk measurement. *Management Science*, 56(10):1833–1848, 2010.
- [15] Martin B. Haugh and Leonid Kogan. Pricing american options: a duality approach. *Operations Research*, 52(2):258–270, 2004.

- [16] Stefan Heinrich. Multilevel Monte Carlo methods. In *Large-scale scientific computing. 3rd international conference, LSSC 2001, Sozopol, Bulgaria, June 6–10, 2001.*, pages 58–67. Springer, 2001.
- [17] Francis A. Longstaff and Eduardo S. Schwartz. Valuing american options by simulation: a simple least-squares approach. *Review of Financial Studies*, 14:113–147, 2001.
- [18] Leonard C. G. Rogers. Monte carlo valuation of american options. *Mathematical Finance*, 12(3):271–286, 2002.
- [19] John N. Tsitsiklis and Benjamin Van Roy. Regression methods for pricing american-style options. *IEEE Transactions on Neural Networks*, 12:694–703, 2001.
- [20] Manuel Villén-Altamirano and José Villén-Altamirano. Analysis of restart simulation: Theoretical basis and sensitivity study. *European Transactions on Telecommunications*, 13(4):373–385, 2002.

FACULTY OF MATHEMATICS, DUISBURG-ESSEN UNIVERSITY, THEA-LEYMANN-STR. 9,
45127 ESSEN, GERMANY

E-mail address: `fabian.dickmann@uni-due.de`

DEPARTMENT OF MATHEMATICS, SAARLAND UNIVERSITY, POSTFACH 151150, 66041
SAARBRÜCKEN, GERMANY

E-mail address: `schweizer@math.uni-sb.de`